Design and Implementation of BcN-NMS

Soonhyang Park, Seonghak Suk, Jaehyung Yoo

Abstract

BcN (Broadband Convergence Network) is a unified network combining data, voice and video IP services. As the BcN emerges, efficient NMS needs to be designed and implemented. In this paper, we propose the method to design and implement the BcN-NMS using several technologies, such as OSS through Java, TMF SID Model and Web Services. We also present the architecture and application structure of BcN-NMS.

Current OSS technology cannot cope with the rapidly increasing scale of networks such as BcN with the diversity of communications technology, and heightened expectations for availability and reliability. In short, service providers need proper OSS solutions nowadays. So, we need a new approach to providing OSS solutions. The OSS/J APIs make up a full OSS solution that supports Service Fulfillment, Assurance, and Billing. In this paper, we used the QoS API and Common API of OSS/J.

We also used WSDL-based XML/SOAP protocol to take much advantage of good compatibility, reduced development period, easy maintenance and low costs in developing BcN-NMS applications.

In addition, we defined the information model of managed elements of BcN-NMS by using the SID model from TMForum. So, the managed information for BcN-NMS is systematically defined by standardized information model, so it can be applicable to various kind of network elements.

The overall proposed scheme can be applied to various domain NMSs. And these structure can improve the reusability of NMSs.
1. Introduction to BcN

BcN (Broadband Convergence Network) is a unified network combining data, voice and video IP services. It contains a number of equipments such as softswitches, access gateways, trunk gateways, signaling gateways, and various kind of servers like application server, media server, number resolution server, and so on.

KT plans to offer a variety of multimedia services such as multimedia ringback, multimedia CID for individual subscribers and multimedia centrex for corporate customers. In addition, there will be various services of video type.

Softswitch is a software-based control switch dedicated to call processing in BcN services.

Gateway systems acts as an intermediate translation mediator between IP packet network and other networks. Access gateway is a gateway system translating the POTS, xDSL and dedicated line traffic into IP packet traffic. Trunk gateway is a gateway system translating the PSTN trunk traffic into IP packet traffic. Signaling gateway is a gateway system translating the No.7 signal into IP packet traffic.

Application server provides application services and value-added services.
Media server performs the real-time processing of a wide variety of media stream.
2. Architecture of BcN-NMS

BcN-NMS consists of network adaptor part, server part and GUI. Server part is classified into four sub-modules such as fault management module, performance management module, configuration management module and service configuration module.

Network adaptor part of BcN-NMS collects the NE information and events through EMS dedicated to each network element. Using the XML/SOAP and WSDL, network adaptor is conveniently collecting the NE information and events from network elements in service/control layer of BcN. In order to collect the information from backbone network of BcN, network adaptor uses SNMP and CLI protocols.

Managed Objects of BcN-NMS are the service/control layer elements such as softswitch, access gateway, trunk gateway, signaling gateway, servers like application server, media server, convergence messaging server, number resolution server, presence and location server. BcN backbone network elements are also managed objects of BcN-NMS.
2.1 Application Architecture of BcN-NMS

BcN-NMS applications are based on J2EE platform and OSS through java APIs. QoS API and Common API of OSS/J are applied to design and implementation of BcN-NMS. And several additional modules such as logging, configuration, exception, and util libraries are implemented.

The OSS through Java Initiative defines a set of APIs, with client access either by tightly or loosely coupled mechanisms, to foster an OSS component market.

The server part of BcN-NMS consists of fault management module, performance management module, configuration management module based on OSS/J QoS and Common APIs.

Each management module make use of J2EE components, JVTSession, ManagedEntity value and Message Driven Bean(MDB). JVTSession controls overall processes, and ManagedEntityValue roles as value object, entity correspondent to database rows.

Events from backend network elements are received and processed by Message Driven Bean(MDB), and then sent to JVTSession which translates and processes the data in proper format. Java Messaging Service(JMS) is used to receive and process events from network elements.

GUI applications call the EJB using RMI/IIOP to directly access the nms server. When server module gives an order to network adaptor, it uses RMI/JRMP. In reverse, the server uses JMS to receive notifications from network adaptor.
2.2 Relation of Application and Server

BcN-NMS takes advantage of OSS/J APIs based on J2EE platform. In this slide, we present the logical view of relations between applications and server. Modules in blue square are outside of NMS server. And modules in violet oval are representing the classes deployed into application server inside of NMS server.

JVTSession is a SessionBean acting as a main server component. It makes use of DAO (Data Access Object) to connect and manipulate DB. DAO can be defined one per DB table. And It has functions like Create, Retrieve, Update, Delete. Many JVTSessions can simultaneously access the DAO, so DAO functions may not be modified for a particular JVTSessions. In that case, we would rather make new DAO for new functions.

JVTSession also uses connectors to communicate with outside of NMS server modules and uses JMS topic to notify data to GUI. To receive data from network adaptor, JVTSession uses JMS Queues.

This structure can improve the reusability and reliability of network management systems.
2.3 Inheritance from TMForum SID Model

SID addresses the information and communication service industry’s need for shared information/data definitions and models. The definitions focus on business entity definitions and attribute definitions. A business entity is a thing of interest to the business, while attributes are facts that further describe the entity. Together the definitions provide a business-oriented perspective of the information and data. Combined with business oriented UML class models, the definitions provide the business view of the information and data. Design oriented UML class models and sequence diagrams provide a system view of the information and data.

TMForum SID (Shared Information Data) consists of six domain packages such as Common Business Entities domain, Resource domain, Service domain, Enterprise domain, Market_Sales Domain and Product Domain.

Among SID domain packages, we inherited several classes from two domain packages, Common Business Entities domain and Resource domain.

Classes extended from Common Business Entities domain are as follows:
- RootEntity, Entity, ManagedEntity

Classes extended from Resource domain are as follows:
- Resource, PhysicalResource, LogicalResource
- PhysicalDevice, Hardware, ManagedHardware
- PhysicalContainer, PhysicalComponent
- Equipment, EquipmentHolder, Card, SystemCard, NetworkCard
2.4 Data Processing example

In the picture, we will show an example process of fault monitoring module of BcN-NMS.

1) Network Adaptor sends the collected fault data to JMS Queue.
2) MDB(Message Driven Bean) of NMS server receives and processes the fault data.
3) NMS server creates the AlarmValue instance.
4) NMS server saves the processed information into DB using DAO(Data Access Object)
5) NMS server creates an AlarmEvent using AlarmEventFactory.
6) 7) Using NotifyProcessor, NMS server sends the information to JMS Topic
8) Subscribing the JMS Topic, GUI can get the desired notification from the JMS Topic.
2.5 Network Adaptor of BcN-NMS

Network adaptor plays an important role in communication with network elements in backbone layer devices and EMSs in service/control layer of BcN. Network adaptor consists of auto-discovery module, performance scheduler, ping scheduler and event handler. In addition, it has information model for network element base on TMF SID.

It also maintains interfaces for communicating NMS and network elements. The NMS-sided interface is called Generic Interface, which provides RMI for on-demand schedule request or synchronous request from/to NMS and JMS for notifying events from network elements to NMS. Receiving the request from NMS, network adaptor send Acknowledge to NMS through RMI interface. Each management class of the network adaptor performs the request, and then send the result to NMS through JMS interface.

The NE-sided interface is called Protocol Service Interface, which can process various protocols such as XML/SOAP, Java ORB, SNMP and CLI. between Network Adaptor and NEs.

Network adapter uses the SNMP or CLI protocol for backbone network layer elements and XML/SOAP protocol for uniform access to service/control network layer elements.
2.6 XML-SOAP (WSDL) Connection

BcN-NMS has to collect various NE information or notified by various NEs. So we need to have a uniform interoperable protocols like XML-SOAP.

XML (eXtensible Markup Language) is used as the message format, and also used as the way to define NE information for FCAPS.

WSDL (Web Service Description Language) is used to define where is the service, what the service can do and how to use the service. In other words, WSDL defines three major elements as follows:

- Defines the structure and the content of messages
- Defines the operations performed on the messages
- Defines the service binding information like port, service at network transport layer for exchanging messages.

First, EMSs deploys appropriate WSDL to private-UDDI.

Second, network adaptor of BcN-NMS searches the UDDI.

Third, if desired service is found, network adaptor downloads WSDL.

Fourth, network adaptors of NMS try to bind to EMSs based on the information in WSDL. And then network adaptors of NMS connect to EMSs via XML over SOAP/HTTP.

In this way, we can easily describe the messaging interface specification between NMS and EMSs. And we also can easily access and use the NE data.
2.7 Results of Implementation (example)

This slide shows the example views of fault monitoring and performance monitoring GUIs of BcN-NMS.

We implemented fault monitoring and performance monitoring modules of BcN-NMS using the technologies described previous slides.

Our fault monitoring part monitors status of network elements like softswitches, gateways, router, applications and servers.

Status informations displayed on the map screen, so that the operators of NMS can monitors the current fault status of BcN and solve the problem in proper way.

In addition, we can see pop-up windows showing detailed information of each faults or network element.

Performance monitoring part monitors and analyzes IP traffic, call level traffic and system performance parameters. It can report the TCA (Threshold Crossing Alarm), so that the operators of NMS can monitors the current TCA and solve the problem in proper way.
3. Consideration

BcN-OSS/BSS needs to be developed through layered approaches for the purpose of efficient management of services and networks.

Required layers can be classified into three parts such as Service Management Layer, SoIP (Service over IP) Management Layer and Network Management Layer.

In Network Management Layer, there are many networks connected to BcN backbone, so BcN-NMS should maintain the interoperabilities with other NMSs for the purpose of stable operation of whole BcN services and network.

In SoIP Management Layer, there will be IP service-dependent NMSs like Quality Management system, SLA Management system, VPN NMS, VoIP NMS, IP-TV NMS and so on.

In Service Management Layer, there will be pure service-dependent systems like Order Entry system, CRM system, Order Management system, Trouble Ticket Management system, and so on.
4. Conclusion

We have made use of various technologies such as OSS through Java, J2EE, TMF SID to design and implement BcN-NMS.

This structure may improve the reusability and reliability of network management systems. Those methodologies can be applicable to other network management systems.

We will apply BcN-NMS to BcN exhibition task of KT, September 2005.

References